CLAIMS

1. (Currently Amended) A method for determining thread switch points within pipeline execution units of a processor, comprising the steps of:

monitoring instruction processing of a first thread within the pipeline execution units;

in the event of a possible switch point within the pipeline execution units, ~~deactivate~~ deactivating the first thread, or not, based upon a first urgency indicator for the first thread, the first urgency indicator being based upon progress of the first thread within the pipeline execution units.

2. (Currently Amended) A method of claim 1, further comprising deactivating the first thread and activating a second thread based upon a second urgency indicator for the second thread, the second urgency indicator being based upon expected progress of the second thread within the pipeline execution units.

3. (Original) A method of claim 2, further comprising deactivating the second thread, or not, based upon the second urgency indicator for the second thread and in the event of a possible switch point event of the second thread.

4. (Original) A method of claim 3, further comprising activating another thread within the pipeline if the second thread is switched out.

5. (Currently Amended) A method of claim 1, the step of deactivating the first thread comprising deactivating the first thread, or not, based upon the first urgency indicator and upon a second urgency indicator of a second thread, the second urgency indicator being based upon expected progress of the second thread within the pipeline execution units.

6. (Original) A method of claim 1, the step of monitoring comprising utilizing a thread controller coupled with the execution units.

7. (Previously Presented) A method of claim 1, further comprising modifying the first urgency indicator to increase or alternatively decrease urgency of the first thread based upon characteristics associated with the possible switch point.

8. (Original) A method of claim 7, further comprising determining whether a time slice expiration occurred.

9. (Original) A method of claim 8, further comprising utilizing a time slice expiration unit.

10. (Original) A method of claim 7, further comprising determining whether a cache miss occurred.

11. (Original) A method of claim 7, further comprising inserting an instruction to the pipeline to change urgency of the thread.

12. (Original) A method of claim 1, further comprising the steps of deactivating the first thread and activating a second thread, and modifying urgency of the second thread.

13. (Currently Amended) A method of claim 1, further comprising the steps of monitoring possible switch points of an inactive thread having a second urgency indicator that is based upon expected progress of the inactive thread within the pipeline execution units, and deactivating the first thread, or not, based upon a first and second urgenciesy indicators.

14. (Currently Amended) A processor for processing multi-threaded program instructions, comprising:

an array of pipeline execution units and associated heuristics affecting how the instructions are processed within the units; and

a thread controller for monitoring processing of the instructions within the units and for switching between multiple program threads based upon (a) the heuristics and (b) urgencies of the program threads;

wherein the urgencies are based upon one or both of (a) progress of the threads through the pipeline execution units and (b) expected progress of the program threads through the pipeline execution units.

15. (Currently Amended) A ~~system~~ processor of claim 14, the heuristics comprising one or more of time slice expiration heuristics, cache miss heuristics and processor interrupt heuristics.

16. (Currently Amended) A ~~system~~ processor of claim 14, the program threads comprising one or more instructions, one of the instructions changing urgency for at least one thread of the processor.

17. (Currently Amended) A ~~system~~ processor of claim 14, the controller modifying an urgency of any of the threads to modify future treatment of the threads in switch out events.

18. (Currently Amended) A ~~system~~ processor of claim 17, the controller either decreasing or increasing urgency for the program threads by injecting an instruction to the pipeline execution units.

19. (Currently Amended) A ~~system~~ processor of claim 14, further comprising a time slice expiration unit for monitoring expiration of threads within the processor.